

PPPPPPPPPPPP		AAAAAAAAAA		SSSSSSSSSSSS		RRRRRRRRRRRR		TTTTTTTTTTTTTT		LLL
PPPPPPPPPPPP		AAAAAAAAAA		SSSSSSSSSSSS		RRRRRRRRRRRR		TTTTTTTTTTTTTT		LLL
PPPPPPPPPPPP		AAAAAAAAAA		SSSSSSSSSSSS		RRRRRRRRRRRR		TTTTTTTTTTTTTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPPPPPPPPPPP		AAA	AAA	SSSSSSSSSS		RRRRRRRRRRRR		TTT		LLL
PPPPPPPPPPPP		AAA	AAA	SSSSSSSSSS		RRRRRRRRRRRR		TTT		LLL
PPPPPPPPPPPP		AAA	AAA	SSSSSSSSSS		RRRRRRRRRRRR		TTT		LLL
PPP		AAAAAAAAAAAAAAAA		SSS		RRR	RRR	TTT		LLL
PPP		AAAAAAAAAAAAAAAA		SSS		RRR	RRR	TTT		LLL
PPP		AAAAAAAAAAAAAAAA		SSS		RRR	RRR	TTT		LLL
PPP		AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP		AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP		AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP		AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP		AAA	AAA	SSSSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLLLLLL
PPP		AAA	AAA	SSSSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLLLLLL
PPP		AAA	AAA	SSSSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLLLLLL

```

PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR      IIIIII  SSSSSSSS  TTTTTTTTTT  RRRRRRRR
PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR      IIIIII  SSSSSSSS  TTTTTTTTTT  RRRRRRRR
PP      PP  AA      AA  SS      WW      WW  RR      RR      II      SS      TT      RR      RR
PP      PP  AA      AA  SS      WW      WW  RR      RR      II      SS      TT      RR      RR
PP      PP  AA      AA  SS      WW      WW  RR      RR      II      SS      TT      RR      RR
PP      PP  AA      AA  SS      WW      WW  RR      RR      II      SS      TT      RR      RR
PPPPPPPP      AA      AA  SSSSSS  WW      WW  RRRRRRRR      II      SSSSSS  TT      RRRRRRRR
PPPPPPPP      AA      AA  SSSSSS  WW      WW  RRRRRRRR      II      SSSSSS  TT      RRRRRRRR
PP      AAAAAAAAAA      SS      WW      WW  WW  RR      RR      II      SS      TT      RR      RR
PP      AAAAAAAAAA      SS      WW      WW  WW  RR      RR      II      SS      TT      RR      RR
PP      AA      AA  SS      WWW  WWW  RR      RR      II      SS      TT      RR      RR
PP      AA      AA  SS      WWW  WWW  RR      RR      II      SS      TT      RR      RR
PP      AA      AA  SSSSSSSS  WW      WW  RR      RR      IIIIII  SSSSSSSS  TT      RR      RR
PP      AA      AA  SSSSSSSS  WW      WW  RR      RR      IIIIII  SSSSSSSS  TT      RR      RR

```

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```



```
1 0001 0 MODULE PASSWRITE_STRING ( %TITLE 'Write a fixed-length string'
2 0002 0 IDENT = '1-002' ! File: PASWRISTR.B32 Edit: SBL1002
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: Pascal Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains a procedure which writes a fixed-length string
36 0036 1 to a textfile.
37 0037 1
38 0038 1 ENVIRONMENT: User mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. SBL 1-April-1981
45 0045 1 1-002 - Make total-width a longword. SBL 30-June-1982
46 0046 1 --
47 0047 1
```

PASSWRITE_STRIN Write a fixed-length string
1-002 Declarations

G 10
16-Sep-1984 02:25:56
14-Sep-1984 12:52:08

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRISTR.B32;1

Page 2
(2)

```

: 49      0048 1 %SBTTL 'Declarations'
: 50      0049 1
: 51      0050 1 PROLOGUE DEFINITIONS:
: 52      0051 1
: 53      0052 1
: 54      0053 1 REQUIRE 'RTLIN:PASPROLOG';
: 55      0117 1 ! Externals, linkages, PSECTs, structures
: 56      0118 1
: 57      0119 1 TABLE OF CONTENTS:
: 58      0120 1
: 59      0121 1
: 60      0122 1 FORWARD ROUTINE
: 61      0123 1 PASSWRITE_STRING: NOVALUE,
: 62      0124 1 PASSWRITEV_STRING: NOVALUE;
: 63      0125 1 ! Write string to textfile
: 64      0126 1 ! Write string to string
: 65      0127 1
: 66      0128 1 MACROS:
: 67      0129 1 NONE
: 68      0130 1
: 69      0131 1 EQUATED SYMBOLS:
: 70      0132 1
: 71      0133 1 NONE
: 72      0134 1
: 73      0135 1 FIELDS:
: 74      0136 1
: 75      0137 1 NONE
: 76      0138 1
: 77      0139 1 OWN STORAGE:
: 78      0140 1
: 79      0141 1 NONE
: 80      0142 1

```



```

82 0143 1 %SBTTL 'PASSWRITE_STRING - Write a fixed-length string to textfile'
83 0144 1 GLOBAL ROUTINE PASSWRITE_STRING (
84 0145 1     PFV: REF $PASSPFV_FILE_VARIABLE,      ! File variable
85 0146 1     STRING_LENGTH: WORD,                ! Length of string
86 0147 1     STRING,                             ! Address of string
87 0148 1     ERROR,                             ! Error unwind address
88 0149 1     TOTAL_WIDTH: SIGNED                 ! Total field width
89 0150 1 ): NOVALUE =
90 0151 1
91 0152 1 ++
92 0153 1 FUNCTIONAL DESCRIPTION:
93 0154 1     This procedure writes a fixed-length string to the specified textfile.
94 0155 1
95 0156 1 CALLING SEQUENCE:
96 0157 1     CALL PASSWRITE_STRING (PFV.mr.r, STRING_LENGTH.rw.v, STRING.rt.r
97 0158 1     [, [ERROR.]r] [, TOTAL_WIDTH.rl.v])
98 0159 1
99 0160 1 FORMAL PARAMETERS:
100 0161 1
101 0162 1     PFV - The Pascal File Variable (PFV) passed by reference.
102 0163 1           The structure of the PFV is defined in PASPFV.REQ.
103 0164 1
104 0165 1     STRING_LENGTH - The length of the string to write
105 0166 1
106 0167 1     STRING - The address of the string to write
107 0168 1
108 0169 1     ERROR - Optional. If specified, the address to unwind to
109 0170 1           in case of an error.
110 0171 1
111 0172 1     TOTAL_WIDTH - Optional. Total field width, defaults to string length.
112 0173 1
113 0174 1 IMPLICIT INPUTS:
114 0175 1
115 0176 1     NONE
116 0177 1
117 0178 1 IMPLICIT OUTPUTS:
118 0179 1
119 0180 1     NONE
120 0181 1
121 0182 1 ROUTINE VALUE:
122 0183 1
123 0184 1     NONE
124 0185 1
125 0186 1 SIDE EFFECTS:
126 0187 1
127 0188 1     If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
128 0189 1
129 0190 1 SIGNALLED ERRORS:
130 0191 1
131 0192 1     LINTOOLON - line too long
132 0193 1     NEGWIDDIG - negative Width or Digits specification is not allowed
133 0194 1
134 0195 1
135 0196 1
136 0197 1 --
137 0198 1
138 0199 2 BEGIN
  
```

PASSWRITE_STRIN Write a fixed-length string
1-002

PASSWRITE_STRING - Write a fixed-length string

I 10
16-Sep-1984 02:25:56
14-Sep-1984 12:52:08

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRISTR.B32;1

Page 4
(3)

```

139      0200      2
140      0201      2
141      0202      2
142      0203      2
143      0204      2
144      0205      2
145      0206      2
146      0207      2
147      0208      2
148      0209      2
149      0210      2
150      0211      2
151      0212      2
152      0213      2
153      0214      2
154      0215      2
155      0216      2
156      0217      2
157      0218      2
158      0219      2
159      0220      2
160      0221      2
161      0222      2
162      0223      2
163      0224      2
164      0225      2
165      0226      2
166      0227      2
167      0228      2
168      0229      2
169      0230      2
170      0231      2
171      0232      2
172      0233      2
173      0234      2
174      0235      2
175      0236      2
176      0237      2
177      0238      2
178      0239      2
179      0240      2
180      0241      2
181      0242      2
182      0243      2
183      0244      2
184      0245      2
185      0246      2
186      0247      2
187      0248      2
188      0249      2
189      0250      2
190      0251      2
191      0252      2
192      0253      2
193      0254      2
194      0255      2
195      0256      2

LOCAL
FCB: REF $PASSFCB_CONTROL_BLOCK, ! File Control block
FIELD_WIDTH, ! Total width of field
PFV_ADDR: VOLATILE, ! Enable argument
UNWIND_ACT: VOLATILE, ! Enable argument
ERROR_ADDR: VOLATILE; ! Enable argument

BUILTIN
ACTUALCOUNT; ! Count of arguments

ENABLE
PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler

!+
!- Get ERROR parameter, if present.
!-

IF ACTUALCOUNT () GEQU 4
THEN
ERROR_ADDR = .ERROR; ! Set unwind address

PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address

!+
!- Validate PFV and get PFV.
!-

PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);

!+
!- Set unwind action to unlock file.
!-

UNWIND_ACT = PASS$K_UNWIND_UNLOCK;

!+
!- Do common initialization.
!-

PASS$INIT_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);

!+
!- See if we have enough room in the record.
!-

IF ACTUALCOUNT () GEQU 5
THEN
BEGIN
FIELD_WIDTH = .TOTAL_WIDTH;
IF .FIELD_WIDTH LSS 0
THEN
$PASS$IO_ERROR (PASS$_NEGWIDDIG,0);
END
ELSE
FIELD_WIDTH = .STRING_LENGTH;
```


PASSWRITE_STRING Write a fixed-length string
1-002

PASSWRITE_STRING - Write a fixed-length string

J 10
16-Sep-1984 02:25:56
14-Sep-1984 12:52:08

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRISTR.B32;1

Page 5
(3)

```

196      0257      3      BEGIN
197      0258      LOCAL
198      0259      EXTRA;
199      0260      EXTRA = (.FCB [FCB$A_RECORD_CUR] + .FIELD_WIDTH) - .FCB [FCB$A_RECORD_END];
200      0261      IF .EXTRA GTR 0
201      0262      THEN
202      0263      $PASSIO_ERROR (PASS_LINTOOLON,1,.EXTRA);
203      0264      END;
204      0265
205      0266      !+
206      0267      ! Move leading blanks, if any
207      0268      !-
208      0269
209      0270      IF .FIELD_WIDTH - .STRING_LENGTH GTR 0
210      0271      THEN
211      0272      BEGIN
212      0273      FCB [FCB$A_RECORD_CUR] = CH$FILL (%C' ', .FIELD_WIDTH - .STRING_LENGTH,
213      0274      .FCB [FCB$A_RECORD_CUR]);
214      0275      FIELD_WIDTH = .STRING_LENGTH;
215      0276      END;
216      0277
217      0278      !+
218      0279      ! Move string to buffer
219      0280      !-
220      0281
221      0282      FCB [FCB$A_RECORD_CUR] = CH$MOVE (.FIELD_WIDTH, .STRING, .FCB [FCB$A_RECORD_CUR]);
222      0283
223      0284      !+
224      0285      ! Call WRITE epilogue routine to move the last character written to the
225      0286      ! user's buffer and to unlock the file variable.
226      0287      !-
227      0288
228      0289      PASS$END_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
229      0290
230      0291      RETURN;
231      0292
232      0293      1      END;
```

! End of routine PASSWRITE_STRING

.TITLE PASSWRITE_STRING Write a fixed-length string
.IDENT \1-002\

.EXTRN PASSWRITE_STRING
.EXTRN PASSWRITE_STRING
.EXTRN PASS\$IO_HANDLER
.EXTRN PASS\$VALIDATE_PFV
.EXTRN PASS\$INIT_WRITE
.EXTRN PASS\$SIGNAL, PASSK_NEGWIDDIG
.EXTRN PASSK_LINTOOLON
.EXTRN PASS\$END_WRITE

.PSECT _PASSCODE, NOWRT, SHR, PIC, 2

.ENTRY PASSWRITE_STRING, Save R2,R3,R4,R5,R6,R7,- ; 0144
R8,R9
MOVAB PASS\$SIGNAL, R9
SUBL2 #8, SP

03FC 00000

59 00000000G 00 9E 00002
5E 08 C2 00009

PASSWRITE_STRIN Write a fixed-length string
1-002

PASSWRITE_STRING - Write a fixed-length string

K 10

16-Sep-1984 02:25:56

14-Sep-1984 12:52:08

VAX-11 Bliss-32 V4.0-742

[PASRTL.SRC]PASWRISTR.B32;1

Page 6
(3)

			7E	D4	0000C	CLRL	ERROR_ADDR	0199
		04	AE	7C	0000E	CLRQ	UNWIND_ACT	
	6D	0082	CF	DE	00011	MOVAL	6\$, (FP)	
	04		6C	91	00016	CMPB	(AP), #4	0218
			04	1F	00019	BLSSU	1\$	
	6E	10	AC	D0	0001B	MOVL	ERROR, ERROR_ADDR	0220
	56	04	AC	D0	0001F	MOVL	PFV, R6	0222
08	AE		56	D0	00023	MOVL	R6, PFV_ADDR	
		00000000G	00	16	00027	JSB	PASS\$VACIDATE, PFV	0228
04	AE		01	D0	0002D	MOVL	#1, UNWIND_ACT	0234
		00000000G	00	16	00031	JSB	PASS\$INIT_WRITE	0240
	05		6C	91	00037	CMPB	(AP), #5	0246
			10	1F	0003A	BLSSU	2\$	
	58	14	AC	D0	0003C	MOVL	TOTAL_WIDTH, FIELD_WIDTH	0249
			0E	18	00040	BGEQ	3\$	0250
			7E	D4	00042	CLRL	-(SP)	0252
	7E	00G	8F	9A	00044	MOVZBL	#PASS\$K, NEGWIDDIG, -(SP)	
	69		02	FB	00048	CALLS	#2, PASS\$\$SIGNAL	
			04	0004B	RET			
	58	08	AC	3C	0004C	MOVZWL	STRING_LENGTH, FIELD_WIDTH	0255
	58	EC	A7	C1	00050	ADDL3	-20(FCB), FIELD_WIDTH, R0	0260
	50	FO	A7	C2	00055	SUBL2	-16(FCB), EXTRA	
			0C	15	00059	BLEQ	4\$	0261
			50	DD	0005B	PUSHL	EXTRA	0263
			01	DD	0005D	PUSHL	#1	
	7E	00G	8F	9A	0005F	MOVZBL	#PASS\$K, LINTOOLON, -(SP)	
	69		03	FB	00063	CALLS	#3, PASS\$\$SIGNAL	
			04	00066	RET			
58	08	AC	10	00	ED	CMPZV	#0, #16, STRING_LENGTH, FIELD_WIDTH	0270
			17	18	0006D	BGEQ	5\$	
	50	08	AC	3C	0006F	MOVZWL	STRING_LENGTH, R0	0273
	58	50	50	C3	00073	SUBL3	R0, FIELD_WIDTH, R0	
50	20		6E	00	2C	MOVC5	#0, (SP), -#32, R0, @-20(FCB)	0274
		EC	B7		0007C			
	EC	A7	53	D0	0007E	MOVL	R3, -20(FCB)	
	58	08	AC	3C	00082	MOVZWL	STRING_LENGTH, FIELD_WIDTH	0275
	EC	BC	58	28	00086	MOVC3	FIELD_WIDTH, @STRING, @-20(FCB)	0282
	EC	A7	53	D0	0008C	MOVL	R3, -20(FCB)	
		00000000G	00	16	00090	JSB	PASS\$END_WRITE	0289
			04	00096	RET			0293
			0000	00097	6\$:	.WORD	Save nothing	0199
	50	08	AC	D0	00099	MOVL	8(AP), R0	
	50	04	A0	D0	0009D	MOVL	4(R0), R0	
		F4	A0	9F	000A1	PUSHAB	ERROR_ADDR	
		F8	A0	9F	000A4	PUSHAB	UNWIND_ACT	
		FC	A0	9F	000A7	PUSHAB	PFV_ADDR	
			03	DD	000AA	PUSHL	#3	
			5E	DD	000AC	PUSHL	SP	
	7E	04	AC	7D	000AE	MOVQ	4(AP), -(SP)	
	00		03	FB	000B2	CALLS	#3, PASS\$IO_HANDLER	
			04	000B9	RET			

; Routine Size: 186 bytes, Routine Base: _PASS\$CODE + 0000

; 233 0294 1
; 234 0295 1 !<BLF/PAGE>


```

236 0296 1 XSBTTL 'PASSWRITEV_STRING - Write string to string'
237 0297 1 GLOBAL ROUTINE PASSWRITEV_STRING (
238 0298 1     MAX_LENGTH: WORD,                                ! Maximum length of string
239 0299 1     STRING_LINE: REF VECTOR [, WORD],             ! String to write to
240 0300 1     STRING_LENGTH: WORD,                          ! Length of STRING
241 0301 1     STRING: REF VECTOR [, WORD],                  ! String to write
242 0302 1     ERROR,                                       ! Error unwind address
243 0303 1     TOTAL_WIDTH: SIGNED                          ! Total field width
244 0304 1 ) : NOVALUE =
245 0305 1
246 0306 1 ++
247 0307 1 FUNCTIONAL DESCRIPTION:
248 0308 1
249 0309 1     This procedure writes a string to the specified string.
250 0310 1
251 0311 1 CALLING SEQUENCE:
252 0312 1
253 0313 1     CALL PASSWRITEV_STRING (MAX_LENGTH.rw.v, STRING_LINE.wvt.r,
254 0314 1     STRING_LENGTH.rw.r, STRING.rt.r [, ERROR.j.r] [, TOTAL_WIDTH.rl.v])
255 0315 1
256 0316 1 FORMAL PARAMETERS:
257 0317 1
258 0318 1     MAX_LENGTH      - The maximum length of STRING_LINE.
259 0319 1
260 0320 1     STRING_LINE   - A varying string to which the output will be appended.
261 0321 1
262 0322 1     STRING_LENGTH  - The length of STRING.
263 0323 1
264 0324 1     STRING        - The string to write.
265 0325 1
266 0326 1     ERROR         - Optional. If specified, the address to unwind to
267 0327 1                   in case of an error.
268 0328 1
269 0329 1     TOTAL_WIDTH    - The width of the field to write. Optional, defaults
270 0330 1                   to STRING_LENGTH.
271 0331 1
272 0332 1 IMPLICIT INPUTS:
273 0333 1
274 0334 1     NONE
275 0335 1
276 0336 1 IMPLICIT OUTPUTS:
277 0337 1
278 0338 1     NONE
279 0339 1
280 0340 1 ROUTINE VALUE:
281 0341 1
282 0342 1     NONE
283 0343 1
284 0344 1 SIDE EFFECTS:
285 0345 1
286 0346 1     NONE
287 0347 1
288 0348 1 SIGNALLED ERRORS:
289 0349 1
290 0350 1     See PASSWRITE_STRING
291 0351 1
292 0352 1 --

```

```

293      0353 1
294      0354 2 BEGIN
295      0355 2
296      0356 2 LOCAL
297      0357 2     PFV: $PASSPFV_FILE_VARIABLE,      ! Pascal File Variable
298      0358 2     ARG_LIST: VECTOR [8, LONG],        ! Argument list
299      0359 2     PFV_ADDR: VOLATILE,                ! Enable argument
300      0360 2     UNWIND_ACT: VOLATILE,              ! Enable argument
301      0361 2     ERROR_ADDR: VOLATILE;              ! Enable argument
302      0362 2
303      0363 2 BUILTIN
304      0364 2     ACTUALCOUNT;                      ! Count of arguments
305      0365 2
306      0366 2 ENABLE
307      0367 2     PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);    ! Enable error handler
308      0368 2
309      0369 2 !+
310      0370 2 ! Get ERROR parameter, if present.
311      0371 2 !-
312      0372 2
313      0373 2 IF ACTUALCOUNT () GEQU 5
314      0374 2 THEN
315      0375 2     ERROR_ADDR = .ERROR;                ! Set unwind address
316      0376 2
317      0377 2 PFV_ADDR = PFV [PFV$R_PFV];              ! Set PFV address
318      0378 2
319      0379 2 !+
320      0380 2 ! Set up ARG_LIST.
321      0381 2 !-
322      0382 2
323      0383 2 ARG_LIST [0] = 3;                        ! Three arguments
324      0384 2 ARG_LIST [1] = PFV [PFV$R_PFV];        ! PFV address
325      0385 2 ARG_LIST [2] = .STRING_LENGTH;         ! Length of STRING
326      0386 2 ARG_LIST [3] = STRING [0];             ! String to write
327      0387 2 IF ACTUALCOUNT () GEQU 6
328      0388 2 THEN
329      0389 2     BEGIN
330      0390 2         ARG_LIST [0] = 5;                ! Add two more arguments
331      0391 2         ARG_LIST [4] = 0;                ! Error address
332      0392 2         ARG_LIST [5] = .TOTAL_WIDTH;    ! Field width
333      0393 2     END;
334      0394 2
335      0395 2 !+
336      0396 2 ! Call PASS$DO_WRITEV to do the work, giving it the address of
337      0397 2 ! PASSWRITE_STRING to call.
338      0398 2 !-
339      0399 2
340      0400 2 PASS$DO_WRITEV (PFV [PFV$R_PFV], .MAX_LENGTH, STRING_LINE [0], ARG_LIST,
341      0401 2     PASSWRITE_STRING);
342      0402 2
343      0403 2 RETURN;
344      0404 2
345      0405 1 END;

```

! End of routine PASSWRITEV_STRING

.EXTRN PASS\$DO_WRITEV

PASSWRITE_STRIN Write a fixed-length string
 1-002 PASSWRITEV_STRING - Write string to string

N 10
 16-Sep-1984 02:25:56
 14-Sep-1984 12:52:08

VAX-11 Bliss-32 V4.0-742
 [PASRTL.SRC]PASWRISTR.B32;1

Page 9
 (4)

			007C	00000	.ENTRY	PASSWRITEV_STRING, Save R2,R3,R4,R5,R6	: 0297
	5E		30	C2 00002	SUBL2	#48, SP	: 0354
		04	7E	D4 00005	CLRL	ERROR_ADDR	
		004F	AE	7C 00007	CLRQ	UNWIND_ACT	
	6D		CF	DE 0000A	MOVAL	3\$, (FP)	
	05		6C	91 0000F	CMPB	(AP), #5	: 0373
			04	1F 00012	BLSSU	1\$	
	6E	14	AC	D0 00014	MOVL	ERROR, ERROR_ADDR	: 0375
08	AE	24	AE	9E 00018	MOVAB	PFV, PFV_ADDR	: 0377
0C	AE		03	D0 0001D	MOVL	#3, ARG_LIST	: 0383
10	AE	24	AE	9E 00021	MOVAB	PFV, ARG_LIST+4	: 0384
14	AE	0C	AC	3C 00026	MOVZWL	STRING_LENGTH, ARG_LIST+8	: 0385
18	AE	10	AC	D0 0002B	MOVL	STRING, ARG_LIST+12	: 0386
	06		6C	91 00030	CMPB	(AP), #6	: 0387
			0C	1F 00033	BLSSU	2\$	
0C	AE		05	D0 00035	MOVL	#5, ARG_LIST	: 0390
		1C	AE	D4 00039	CLRL	ARG_LIST+16	: 0391
20	AE	18	AC	D0 0003C	MOVL	TOTAL_WIDTH, ARG_LIST+20	: 0392
	55	FF01	CF	9E 00041	MOVAB	PASSWRITE_STRING, R5	: 0400
	54	0C	AE	9E 00046	MOVAB	ARG_LIST, R4	
	56	24	AE	9E 0004A	MOVAB	PFV, R6	
	53	08	AC	D0 0004E	MOVL	STRING_LINE, R3	
	52	04	AC	3C 00052	MOVZWL	MAX_LENGTH, R2	
		00000000G	00	16 00056	JSB	PASS\$DO_WRITEV	
			04	0005C	RET		: 0405
			0000	0005D	.WORD	Save nothing	: 0354
	50	08	AC	D0 0005F	MOVL	8(AP), R0	
	50	04	A0	D0 00063	MOVL	4(R0), R0	
		CC	A0	9F 00067	PUSHAB	ERROR_ADDR	
		D0	A0	9F 0006A	PUSHAB	UNWIND_ACT	
		D4	A0	9F 0006D	PUSHAB	PFV_ADDR	
			03	DD 00070	PUSHL	#3	
			5E	DD 00072	PUSHL	SP	
	7E	04	AC	7D 00074	MOVQ	4(AP), -(SP)	
00000000G	00		03	FB 00078	CALLS	#3, PASS\$IO_HANDLER	
			04	0007F	RET		

; Routine Size: 128 bytes, Routine Base: _PASS\$CODE + 00BA

: 346 0406 1
 : 347 0407 1 !<BLF/PAGE>

PASSWRITE_STRIN Write a fixed-length string
1-002 PASSWRITEV_STRING - Write string to string

B 11
16-Sep-1984 02:25:56 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:52:08 [PASRTL.SRC]PASWRISTR.B32;1

Page 10
(5)

: 349 0408 1 END
: 350 0409 1
: 351 0410 0 ELUDOM

! End of module PASSWRITE_STRING

PSECT SUMMARY

: Name Bytes Attributes
: _PASS\$CODE 314 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	0	0	581	00:01.0
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	96	22	33	00:00.4

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PASWRISTR/OBJ=OBJ\$:PASWRISTR MSRC\$:PASWRISTR/UPDATE=(ENH\$:PASWRISTR
:)

: Size: 314 code + 0 data bytes
: Run Time: 00:07.7
: Elapsed Time: 00:16.9
: Lines/CPU Min: 3178
: Lexemes/CPU-Min: 13875
: Memory Used: 90 pages
: Compilation Complete

0298 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PASWIREH
LIS

PASWIREL
LIS

PASWIREG
LIS

PASWISTR
LIS

PATCH

PATDEF
MDL

PASWIREF
LIS

PATCH
MAP

SRMDEF
MDL

PASWIRFH
LIS

PASWIRAR
LIS

PASWIREL
LIS

PASWIRLS
LIS

PASWIRINT
LIS

PASWIRIOCT
LIS

PASWIRFG
LIS

BSTRUC
REQ

PASWIRFF
LIS

CHRKEY
REQ

PASWIRFD
LIS